**React.js Workshop Setup Instructions**

**Following are the steps for installation of React.js related software on Windows systems.**

Please read the complete instructions and **follow exactly as per the instruction and install exactly in the mentioned folders.**

**Note:    <Drive>:  mentioned below is C: or D: or E: or F: etc. wherever space is available for installation.**

1. Install Node.js v8.12.0 - 32 Bit or 64 Bit depending on your system type. **Admin privilege is required.**
   1. <https://nodejs.org/dist/v8.12.0/node-v8.12.0-x86.msi> [ 32 Bit Installer ]

**(OR)**

* 1. <https://nodejs.org/dist/v8.12.0/node-v8.12.0-x64.msi> [ 64 Bit Installer ]

1. You can use editors like Atom Editor [ <https://atom.io/> ] or Microsoft Visual Studio Code [ <https://code.visualstudio.com> ]
2. Install **Google Chrome**. (Preferred one, though you can use other browsers).
3. Open Google Chrome. **Admin privileges might be required.**
   1. Go to CHROME WEB STORE
   2. Type “**React Developer Tools**” in “Search the store” box.
   3. Click “**Add to Chrome**” on the “React Developer Tools” extension.

**Note:** This step is NOT a mandatory step to run React applications. If you are not able to install from behind proxy, you can ignore it.

1. Create a folder **“fullstack”** under **<Drive>:\.** Let us call this path as **<FS\_PATH>**
2. Create a folder **react\_ws** under **<FS\_PATH>**
3. Goto “**Node.js command prompt**”. (**Use  the shortcut** present under the “Node.js” traversed through “Start | All Programs”.
   1. Change directory to **<FS\_PATH>\react\_ws**
4. **IMPORTANT NOTE:** This step is required only if you are installing behind proxy. **Otherwise, do not issue the below commands.**

If behind proxy, Configuring proxy for node.js is important using npm config. Only if **proxy** is configured, “npm  install” will work properly.

* **npm config set proxy**[**http://ADUserID:ADpassword@proxyname:port**](http://ADUserID:ADpassword@proxyname:port)
* **npm config set https-proxy**[**http://ADUserID:ADpassword@proxyname:port**](http://LDAP-ADUserID:LDAP-ADpassword@proxyname:port)
* **IMPORTANT (OPTIONAL STEP – NOT A REGULAR ONE) :** In case, you have already set the proxy config wrongly and want to run the remaining commands from Direct Connection to Internet, then to delete the proxy configuration, execute the following commands.
  + **npm config delete proxy**
  + **npm config delete https-proxy**

1. Run the following commands

**NOTE** : **devDependencies** are modules which are only required during development, while **dependencies** are modules which are also required at runtime. To save a dependency as a devDependency on installation we need to do an npm install **--save**-**dev** , instead of just an npm install **--save**.

**npm install -g create-react-app**

// install react tool (SIMILAR TO ANGULAR CLI)

**create-react-app APPNAME**

// create new APPNAME (SIMILAR TO ***ng new APPNAME***)

**npm start**

// start REACT app (SIMILAR TO ***ng serve***)

**npm init –y**

// create default package.json

**npm install --save** [**axios@0.18.0**](mailto:axios@0.18.0)

// Promise based HTTP client for the browser and node.js

**npm install --save-dev babel-core**

**npm install --save-dev @babel/core**

// Babel is a JavaScript compiler, to convert ECMAScript 2015+ code into a backwards compatible version of JavaScript in current and older browsers or environments

**npm install --save-dev babel-loader**

// Babel module loader for webpack

**npm install --save-dev** [**babel-preset-env@1.7.0**](mailto:babel-preset-env@1.7.0)

**npm install --save-dev @babel/preset-env**

// A Babel preset for each environment

**npm install --save-dev** [**babel-preset-react@6.24.1**](mailto:babel-preset-react@6.24.1)

**npm install --save-dev @babel/preset-react**

// Babel preset for all React plugins

**npm install --save-dev** [**prop-types@15.6.2**](mailto:prop-types@15.6.2)

// Runtime type checking for React props and similar objects

**npm install --save** [**create-react-class@15.6.3**](mailto:create-react-class@15.6.3)

// Legacy API for creating React components

**npm install --save flux@3.1.3**

// An application architecture for React utilizing a unidirectional data flow.

**npm install --save react-redux**

**//** Similar to Facebook's Flux architecture

**npm install --save redux**

**//**

**npm install --save redux-thunk**

**//**

**npm install --save** [**lodash@4.17.11**](mailto:lodash@4.17.11)

// A modern JavaScript utility library delivering modularity, performance, extras.

**npm install --save** [**object-assign@4.1.1**](mailto:object-assign@4.1.1)

// ES2015 `Object.assign()` ponyfill

**npm install --save react**

//  a JavaScript library for building user interfaces.

**npm install --save** [**react-dom@16.5.2**](mailto:react-dom@16.5.2)

// React package for working with the DOM

**npm install --save** [**react-router-dom@4.3.1**](mailto:react-router-dom@4.3.1)

// DOM bindings for React Router

**npm install --save-dev** [**webpack@4.22.0**](mailto:webpack@4.22.0)

// to bundle JavaScript files for usage in a browser, yet it is also capable of transforming, bundling, or packaging just about any resource or asset.

**npm install --save-dev** [**webpack-cli@3.1.2**](mailto:webpack-cli@3.1.2)

// CLI for webpack & friends

**npm install -g** [**webpack@4.22.0**](mailto:webpack@4.22.0)

//

**npm install -g webpack-cli@3.1.2**

//

**npm install -g http-server@0.11.1**

// A simple zero-configuration command-line http server

**npm install -g lite-server@2.4.0**

// Lightweight development node server for serving a web app, providing a fallback for browser history API, loading in the browser, and injecting scripts on the fly.

1. For one of the demo which involves http requests, install json-server so that http requests could be served.

**npm install –g json-server**